A REAL TIME IMAGE PROCESSING SUBSYSTEM: GEZGIN

N. Ismailoglu, O.Benderli, I. Korkmaz, M. Durna, T. Kolçak, Y. Ç. Tekmen
TÜBITAK-BILTEN
Middle East Technical University, 06531 Ankara, TURKEY
Phone: +90 (312) 210 13 10 Fax: +90 (312) 210 10 55
{neslin,oguz}@vlsi.bilten.metu.edu.tr
{ilgaz.korkmaz, mehmet.durna, taner.kolcak}@bilten.metu.edu.tr
Electrical-Electronics Engineering Dept.,
Middle East Technical University, 06531 Ankara, TURKEY
Phone: +90 (312) 210 23 44
tekmen@metu.edu.tr

Abstract

In this study, a real-time image processing subsystem, GEZGIN, which is currently being developed for BILSAT-1, a 100kg class micro-satellite, is presented. BILSAT-1 is being constructed in accordance with a technology transfer agreement between TÜBITAK-BILTEN (Turkey) and SSTL (UK) and planned to be placed into a 650 km sun-synchronous orbit in Summer 2003. GEZGIN is one of the two Turkish R&D payloads to be hosted on BILSAT-1. One of the missions of BILSAT-1 is constructing a Digital Elevation Model of Turkey using both multi-spectral and panchromatic imagers. Due to limited down-link bandwidth and on-board storage capacity, employment of a real-time image compression scheme is highly advantageous for the mission. GEZGIN has evolved as an implementation to achieve image compression tasks that would lead to an efficient utilization of both the down-link and on-board storage. The image processing on GEZGIN includes capturing of 4-band multi-spectral images of size 2048x2048 8-bit pixels, compressing them simultaneously with the new industry standard JPEG2000 algorithm and forwarding the compressed multi-spectral image to Solid State Data Recorders (SSDR) of BILSAT-1 for storage and down-link transmission. The mission definition together with orbital parameters impose a 6.5 seconds constraint on real-time image compression. GEZGIN meets this constraint by exploiting the parallelism among image processing units and assigning compute intensive tasks to dedicated hardware. The proposed hardware also allows for full reconfigurability of all processing units.

Introduction

The BILSAT-1 Mission Background

Small satellites for Low Earth Orbits (LEO) provide fast and cheap access to space. For this orbit range, launch opportunities are relatively cheaper and shared launches are possible. These factors make it possible to use Commercial Off The Shelf (COTS) components. Besides, reducing the redundancy level of the platform allows for further cost reduction in non-critical missions. Altogether, the cheaper initial costs make it easier for a company to enter the space arena. Therefore, there is a growing trend in small satellite technologies.

To make a start-up in space technologies in Turkey, TÜBITAK[*]-BILTEN[**] has initiated a micro-satellite project [1]. The micro-satellite, named BILSAT-1, will be orbiting in LEO equipped with necessary payloads serving remote sensing applications. Remote sensing has an abundance of civilian applications in Turkey such as agriculture, tourism, deforestation monitoring, city planning and disaster monitoring. Within the framework of this project; BILSAT-1 is to be designed and built including construction of the entire infrastructure such as clean rooms for assembly and integration, research and development labs, ground station to track and control satellite(s). This project is being carried out in accordance with a technology transfer agreement between TÜBITAK-BILTEN (Turkey) and SSTL (UK), a leading international company in small satellite technologies.

Currently the Know-How Transfer Team (KHTT) of TÜBITAK-BILTEN engineers are accommodated in

[*] TÜBITAK, a non profit governmental organisation of Turkey, is an acronym for The Scientific and Technical Research Council of urkey

[**] BILTEN is an acronym for Information Technologies and Electronics Research Institute and is affiliated with TÜBITAK

SSTL facilities to participate in all phases of design and production of BILSAT-1. By the end of the project, the KHTT team will have acquired the necessary know-how to design and build a small satellite on the premises of TÜBITAK-BILTEN.

To expedite acquisition of the necessary space technology, an additional power and mass budget has been allocated to Turkish R&D payloads, which are designed and manufactured by Turkish engineers on TÜBITAK-BILTEN premises. One of the payloads is an experimental multispectral camera (so called ÇOBAN) with 80m Ground Sample Distance (GSD) in nine channels. The other payload which is presented in detail in this paper is a high-performance real-time image processing system (so called GEZGIN) that is to perform real time JPEG2000 image compression. GEZGIN is a highly integrated and fully reconfigurable signal processing system, which allows full software upgrade in orbit. This flexible architecture provides robustness against failures in orbit and enables addition of other sophisticated on-board image processing features.

Why spend money for a real-time JPEG2000 compression subsystem in space?

For low-cost commercial small satellites in LEO, on-board data storage and efficient utilization of the communication bandwidth are critical. Increasing on-board data storage capacity requires more memory components, increases power consumption, circuit complexity, and cost, while reducing the overall reliability of the system. On the other hand, most of the data associated with the LEO satellites are images of the earth for specific remote sensing purposes. The bandwidth of the communication channel and the link opportunities of the satellite with the ground station constrain the amount of data transfer. Therefore, efficient bandwidth utilization is critical.

For civilian earth imaging applications, a straightforward solution is to utilize a high performance compression technique such as JPEG2000 [2]. On-board compression enables more images to be captured for a given platform and gives provision of coping with the increased image sizes captured by the increased charge coupled devices (CCDs) [3]. In addition, implementing onboard editing algorithms reduces the amount of useful data to be downloaded radically by clipping the useless parts such as clouds [4].

JPEG2000 is the new industry standard for still image compression . Main features of JPEG2000 are:

1. 40%-60% lossless compression
2. No blocking effects at high compression rates
3. Support for multi-spectral imaging
4. Robustness in noisy channels
5. Inherent thumbnail generation allowing progressive image reconstruction: Initial viewing of a low resolution image while transmission is in progress, transmitting the details on demand
6. Region-Of-Interest (ROI) coding
7. Random access code streaming for extraction and reconstruction of data by area-of-interest, resolution, color format and quality

On the other hand, Consultative Committee for Space Data Systems (CCSDS) is closely considering the JPEG2000 standard as a recommendation for being homogeneously used in space systems and their ground archive facilities [5]. In the CCSDS Compression Working Group Sub-panel P1A held in 1998, CCSDS has stated the requirements for lossless and lossy compression techniques. These techniques should be suitable for both frame and non-frame (push broom) data generators, possess a scalable compression rate, accept images with dynamic ranges up to 16 bits/pixel, be capable of processing data-inputs up to 20 pixels/sec at not more than 1 Watt/pixel in real-time. The system should operate with minimum human intervention and should contain data packing for error-resilience [6]. Like CCSDS, European Space Agency (ESA) is strongly interested in a new compression technique, using the same computational technique as JPEG2000 which utilizes a wavelet kernel [7].

Mission Definition and the BILSAT-1 platform

BILSAT-1 is a small satellite based on the enhanced micro-satellite platform designed by SSTL. The satellite is planned to be launched into a 650 km, 10:30AM-10:30PM sun-synchronous orbit. BILSAT-1, unlike many other micro-satellites, has the capability of three-axis attitude control. With this capability, the satellite can perform fast pitch manoeuvres to take stereo images. Considering 650 km orbit, the satellite has a total of 40 minutes of contact time per day on the average with the ground station based in BILTEN.

The main platform consists of three on-board computers based on Intel 186 and 386 processors, two VHF receivers and two UHT transmitter for backing up the primary S-Band receiver and transmitters operating at 2 Mbps. The system is controlled with a high precision attitude determination and control subsystem.

The main imaging system of BILSAT-1 consists of a four-band (Red, green, blue, infra-red), 26m GSD, multispectral imager and a high-resolution, 12m GSD panchromatic imager. Each imager has an output rate of 20 Mbps adding up to a 100 Mbps total throughput of 8

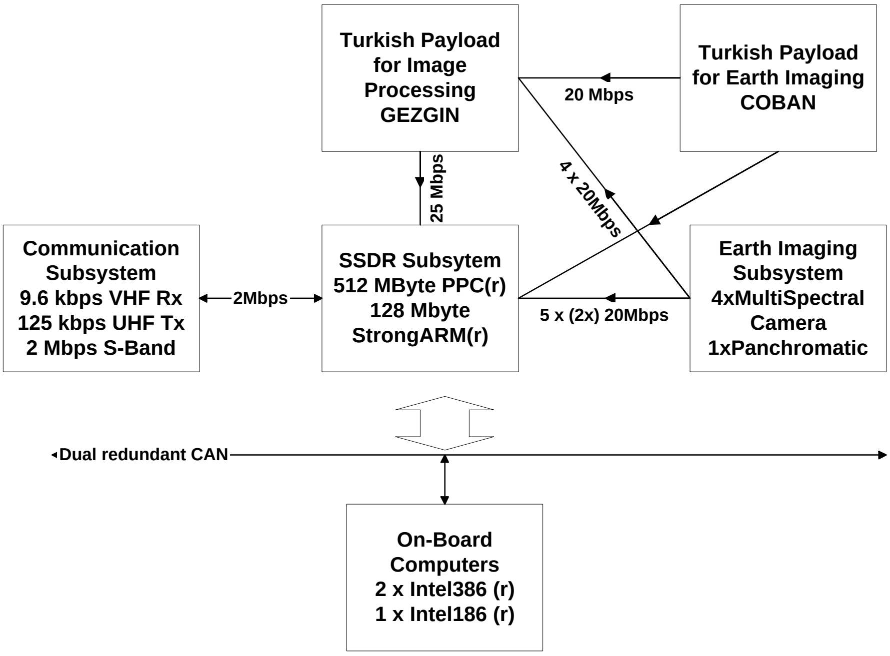


Figure 1. Block Diagram of BILSAT-1

bits/pixel images. The Motorola PPC® and Intel StrongARM®based Solid State Data Recoding units (SSDRs) on BILSAT-1 are equipped with an off-line compression algorithm. However, BILSAT-1 mission has been constrained by the down-link capacity, which is 2 Mbps with a link time of 40 min/day. To increase the number of downloaded images, GEZGIN is added to BILSAT-1 as an R&D payload.

The multi-spectral cameras utilize 2048x2048 pixel COTS CCDs as image capturing elements. The dynamic range of the output is 8 bits/pixel resulting in a 4 Mbyte frame size for each camera. When the cameras are triggered to capture an image, each camera transmits its data to both the primary SSDR and GEZGIN. The block-diagram of BILSAT-1 is given in Figure 1.

The mission of BILSAT-1 is defined as follows:

1. Construction of a Digital Elevation Model (DEM) of Turkey using multi-spectral and panchromatic cameras
2. Capturing images for remote sensing applications
3. Providing images (from the entire region of coverage of BILSAT-1) of disaster areas to the Disaster Monitoring Consortium to which TÜBITAK-BILTEN is a member

DEM generation of Turkey is done by capturing the images of the entire country with 20% overlapping consecutive images. The mission definition together with orbital parameters impose a 6.5 seconds performance constraint for real-time image processing on GEZGIN between two consecutive images with 57x57 km^2 swat. The design problem for GEZGIN, therefore is, to compress image data with a total input bandwidth of 80 Mps and transfer the compressed data to an SSDR over a 25 Mbps link in 6.5 seconds.

General Architecture

GEZGIN achieves its mission by exploiting the parallelism among image processing units and assigning compute intensive tasks to dedicated hardware.

The JPEG2000 algorithm contains many iterative blocks and parallel structures. Breaking out such blocks/structures into multiple data paths and implementing them in reconfigurable hardware improve the overall system performance without loss of flexibility [7]. Simulations on typical satellite images show that 70% of processing time for JPEG2000 compression is spent in wavelet transformation which contains such iterations and parallel structures, and 30%

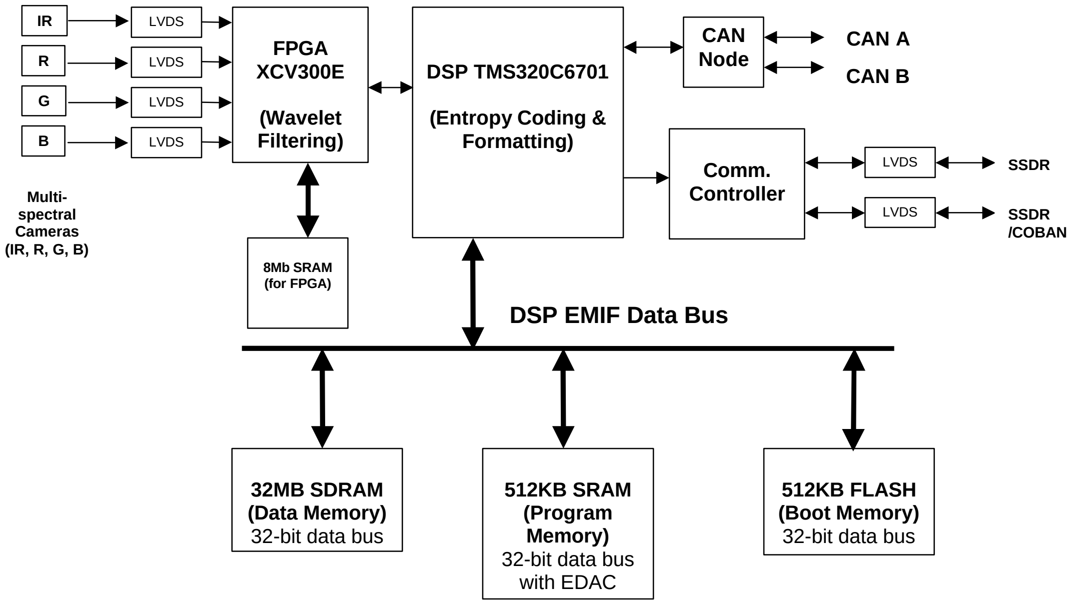


Figure 2. Block Diagram of GEZGIN

is spent in entropy coding and formatting. Therefore, the JPEG2000 compression task on GEZGIN is distributed as follows:

1. Wavelet transformation which consists of 5/3 coefficient integer filtering and signal decomposition, is implemented on a Xilinx Virtex-E type Field Programmable Gate Array (FPGA),

2. Entropy coding and formatting are implemented on a general purpose Digital Signal Processor (DSP), namely TMS320C6701 of TI.

With this distributed processing, GEZGIN attains a high throughput and maintains real-time operation.

The block diagram of GEZGIN is given in Figure 2. The image data is received through four dedicated high speed data links from the infra-red (IR), red (R), green (G) and blue (B) band cameras. Although a dedicated link is provided for each camera, two of the cameras are active at a time (due to SSDR bandwidth requirements), and a complete frame for each band is transmitted synchronously.

Image capturing and wavelet transformation is performed in the FPGA (Xilinx XCV300E). Since two camera channels are active at a time, the FPGA employs two parallel wavelet filtering pipelines. Some buffering of the received image frames is required on off-chip SRAM modules prior to wavelet filtering. As the image frames are filtered, the output is continuously transferred to the external data memory of the DSP (SDRAMs) through the Host Processor Interface (HPI) link between the FPGA and the DSP, operating at the processor clock rate.

The DSP performs entropy coding and compression on the data stored in SDRAMs and transfers the compressed frames through a dedicated Communication Controller IC (CCIC) and high speed data links to SSDRs.

All data links on GEZGIN use the LVDS standard [8] at the physical layer due to speed, power and robustness considerations.

Implementation

Figure 3 shows the JPEG2000 algorithm flow. Only the compressor part of the flow is implemented on GEZGIN.

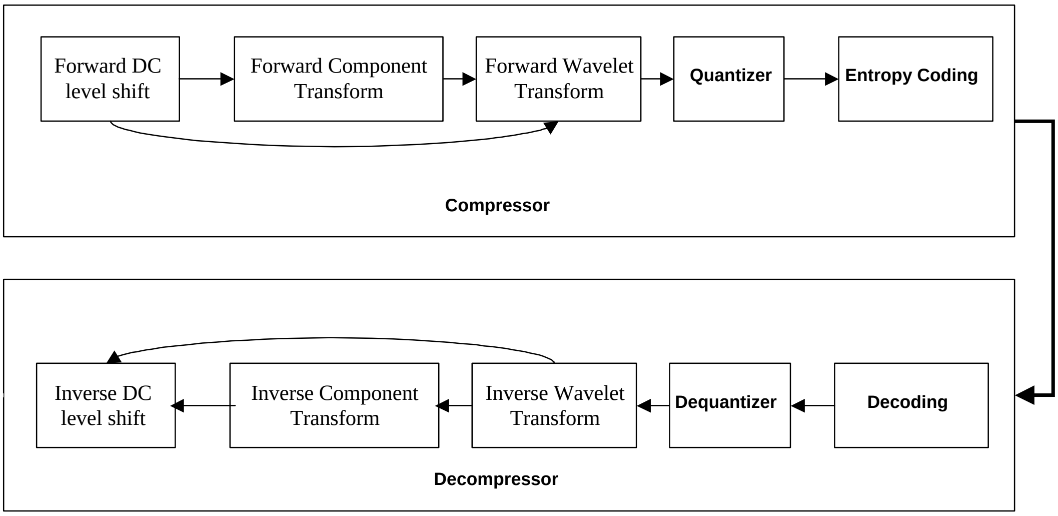


Figure 3. JPEG2000 Algorithm

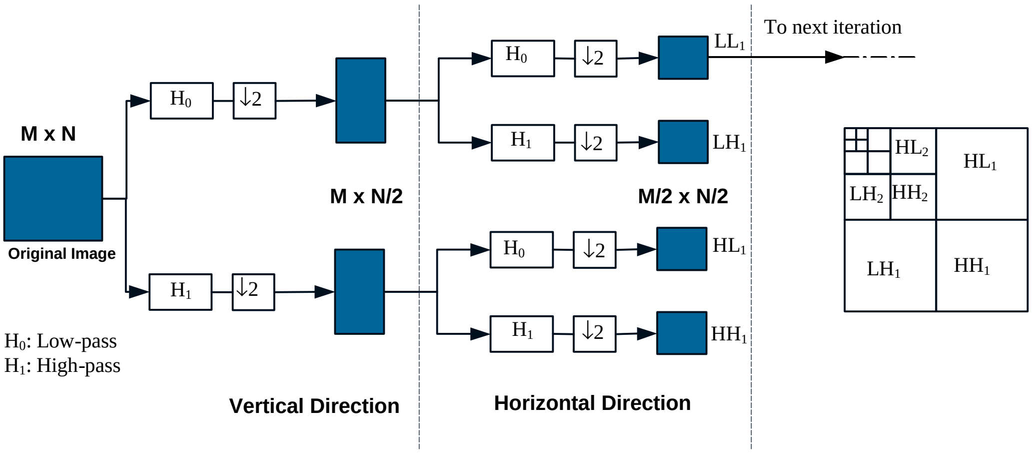


Figure 4. Block Diagram of Discrete Wavelet Transform

Wavelet Transformation

Wavelet based image compression methods like JPEG2000 involve two dimensional wavelet analysis of the image components. Figure 4 shows the block diagram of Discrete Wavelet Transform (DWT) used in JPEG2000. One level of sub-band decomposition consists of low-pass and high-pass filtering of the image, once in vertical and then in horizontal direction. Further sub-band decomposition is possible by applying the same filtering to the LL sub-band. In GEZGIN two level sub-band decomposition is applied.

At each level of filtering the image is decomposed into sub-bands carrying course and detail information. The output of a horizontal high-pass filter carries details such as vertical edges, whereas output of low-pass filter contains the dc and the low frequency information. Figure 5 shows a typical tree-structured sub-band transformation.

Of the two wavelet transformations defined in the JPEG2000 standard, 5/3 filtering is preferred due to its reversibility and suitability for lossless compression. 5/3 filtering is also simpler to implement in hardware, since the transform maps integers to integers, hence

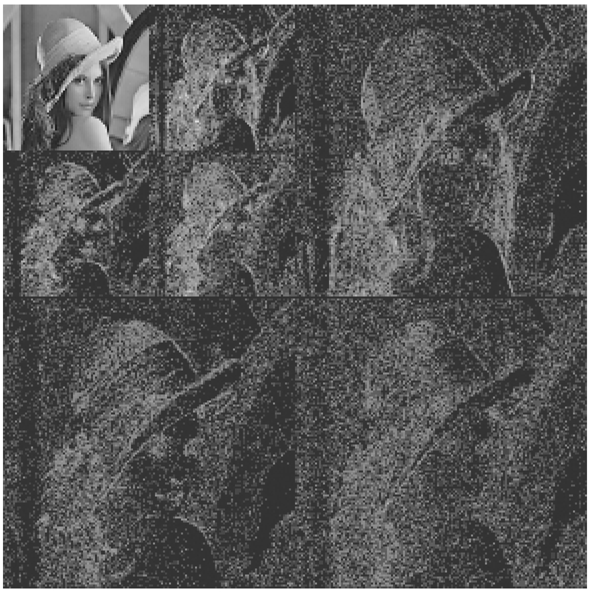


Figure 5. Typical tree-structured sub-band transformation.

reqires no floating-point operations. Besides, the DC gain (K_0) of the 5/3 filter is unity, which prevents expansion of the dynamic ranges after filtering. Hence, no quantization is required after 5/3 filtering [8].

In order to reduce the memory requirements of the architecture, factorization of 5/3 filter into lifting steps is used in hardware implementation [8,9]. Figure 6 shows the lifting implementation of the wavelet filter.

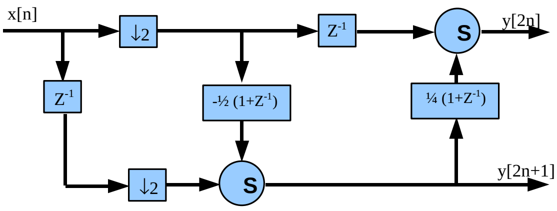


Figure 6. Lifting implementation of the wavelet filter.

The image data is streamed into GEZGIN serially in row-based manner [10], simultaneously from four cameras. Each image frame consists of four 1024x1024 pixel size quadrants as shown in Figure 7. Due to low-memory considerations on configurable logic it is not feasible to buffer lines of length 1024 from four concurrent image streams. Therefore, GEZGIN performs DWT on 64 independent tiles of size 256x256 pixels.

Figure 8 shows the image data flow in the FPGA. The input sequencer takes the serial image data from four channels and writes it to the external SRAM units in

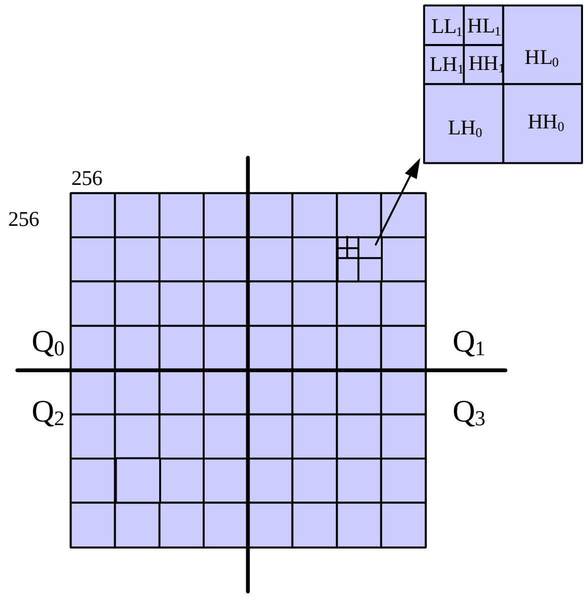


Figure 7. Partitioning of a 2048x2048 frame into 64 tiles of size 256x256. 2 levels of sub-band decomposition is applied to each tile.

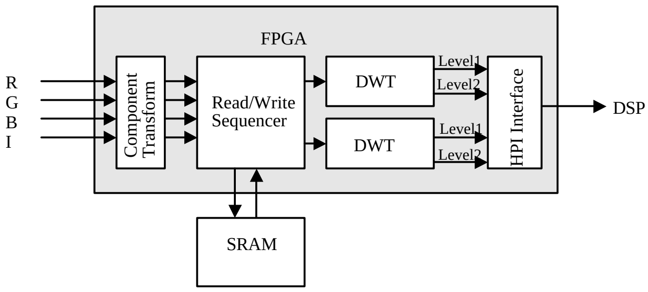


Figure 8. Data Flow in the FPGA.

parallel. As soon as 256 rows of the first quadrant of each channel is buffered, the input sequencer starts reading back the tiles of two channels in rows of 256 pixels. The rows are fed into two parallel filter banks. The storage elements required by the tapped delay line architecture of the filters are implemented in internal RAM blocks of the FPGA. The filters include logic to identify image boundaries for symmetric extension [2,8].

The output from the filter occurs in bursts which would result in a bottleneck on the HPI bus. Therefore, the first level of outputs is delayed with one line depth buffer, so the data bursts from each sub-band are alternated. This buffering scheme can be extended to further levels of filtering.

To reduce computation in the DSP, the dynamic range of each sub-band is determined and embedded in the transformed image data.

Entropy Coding and Formatting

JPEG2000 uses a specific version of arithmetic coding known as MQ coder to generate an embedded representation for each code-block of quantized sub-band samples. After quantization of wavelet coefficients is performed in the encoder, the quantization indices for each sub-band are partitioned into codeblocks. Codeblocks have rectangular shapes and maximum number of any samples in any codeblock must not exceed 4096 with nominal width and height of an integer power of two [11].

Codeblock encoding in MQ coder is essentially a bit-plane encoder. Although bit-plane coding is similar to the coding techniques used in embedded zero-tree wavelet (EZW)12 and set partitioning in hierarchical trees (SPIHT)13, bit-plane coding in MQ coder of JPEG2000 has no inter-band dependencies and uses three passes instead of two. Each codeblock is completely included in one sub-band and coded independently from the other codeblocks which are in the same sub-band. The introduction of the third pass reduces the amount of data associated with each coding pass and allows a finer rate control.

The three coding passes in JPEG2000 MQ coder are significance propagation, magnitude refinement and cleanup pass. Significance pass is used to flag the significant samples in a bit-plane and predict their sign information. Magnitude refinement is used to code the next magnitude bit of a sample which is flagged to be significant. Cleanup pass deals with significance and sign information of insignificant samples. All these coding passes follow a stripe-oriented scan through the codeblock samples. Each stripe represents four rows of codeblock samples and stripes are always aligned with the top of the codeblock. Although all samples are scanned in stripes in these coding passes, information for any sample in codeblock is encoded only in one of the three coding passes for each bit-plane [2].

MQ encoder may be visualized as a state machine, which maps a sequence of input symbols *x Î {0,1}* and their associated context labels *K* to compressed codewords. The codeword segments are generated incrementally and simultaneously with the symbol and context pairs *{x, K}*. Coding pass memberships and context labels *K* are determined within a 3x3 neighbourhood in stripe-oriented scan [11]. During this operation, the following components are used:

1. A set of internal state variables *A, C, t, T,* and *L*. *A* and *C* are the registers common to the arithmetic coding implementations, which denote the probability interval length and probability lower bound for probability mapping respectively. *L* represents the number of code bytes generated and *t* is a counter which triggers the event that partially generated code bits should be moved out of the *C* register to the byte buffer *T*.
2. Context state files for each possible context *K*, consisting of pair entries *(S, s)*. The single bit *s* identifies the most probable symbol for context *K* while *S* identifies the most probable symbol probability estimate.
3. Set of probability mapping rules (tables) to interpret and manipulate context state files associated with the contexts. These lookup tables hold the relationships between the state value S and least probable symbol estimates, new values for S depending on the coded symbol (the coded symbol may be the most or least probable symbol) and MPS-LPS symbol exchange values when the coded symbol is the least probable symbol.

JPEG2000 defines 19 contexts and 4 tables each having 47 entries.

As stated above, the MQ coder is byte oriented, i.e., codeword segments must consist of a whole number of bytes. The MQ coder adopts a bit stuffing approach to avoid the need for full carry resolution. This process ensures that carry bits arising from arithmetic operations on *C* register cannot propagate into byte buffer *T*. With this approach, some combinations of code bytes cannot arise (a byte following 0xFF must be in the range 0x00 through 0x8F). JPEG2000 uses values 0xFF90 through 0xFFFF by assigning them to markers in the encoded file stream.

Lossy versus lossless compression

JPEG2000 lossless compression requires reversible non-linear operations for encoding and decoding, i.e., reversible color transform, reversible DWT (non-linear filtering operations such as 3/5 tap filtering) and no quantization. Though GEZGIN utilizes lossless compression, if desired, some loss can be introduced into the encoding procedure as follows.

1. Quantization of wavelet coefficients: The dynamic range of the wavelet coefficients resulting from reversible DWT is 10 bits (for 8 bit pixels) and is not quantized for lossless compression. However, truncation of these coefficients to fewer bits, e.g. 8, will result in a simple quantization of these coefficients, e.g. to 256 levels for 8 bit wavelet coefficients, and will introduce loss to the compression. However, the decreased dynamic range of

wavelet coefficients will yield less processing time for the MQ coder due to a reduced number of bit-planes and hence coding passes.

2. Omission of sub-band(s) from lower decomposition levels: High frequency components of lower decomposition levels can be omitted from encoding to achieve better compression rates.

Region of Interest Coding

JPEG2000 provides a mechanism where certain regions of the image may be assigned a higher priority. This property is known as ROI coding. ROI coding can use lossless compression for regions of interest and lossy compression for other regions [11].

ROI capability offered by JPEG2000 involves scaling of the important sub-band samples (samples that are in region of interest) by some non-negative integer *U*. *U* denotes the up-shift value and is determined by the dynamic range of the sub-band in interest.

ROI coding also allows the regions of interest to be encoded and decoded first with lossless compression [11].

Performance Results

Performance tests of the JPEG2000 algorithm implemented by GEZGIN are summarized in Table 1. All performance tests were performed with the following settings: 2 level wavelet composition (3/5 reversible tap filter), no color transform, half of sub-band size codeblocks, single tile, and DC level shift of 128. Figure 9 shows samples of original and compressed images.

Memory Subsystem

The memory subsystem of GEZGIN consists of separated data and program memory spaces. The program memory has been designed with triple modular redundancy and hardware Error Detection And Correction (EDAC) to handle Single Event Upsets (SEUs). The EDAC algorithm is implemented in anti-fuse FPGAs as majority voting on three redundant memory units (SRAMs). This scheme is sufficient to handle the typical SEU rate at BILSAT-1's orbit, which is one bit in one million bits per day.

A permanent copy of the bootstrap and program code is stored on-board GEZGIN in non-volatile memory. Depending on the selected operation mode, the bootstrap code copies either the program code from non-volatile memory or a new program code (received through the CAN controller) to the EDAC protected SRAM modules.

The data memory is used as temporary storage for the wavelet transformed images. Since SEUs will manifest only as image noise at this stage, no EDAC protection is necessary for the data memory. Access to the data memory modules (SDRAMs) is shared by the FPGA and the DSP through the HPI.

Data Link Interface

The data flow in and out of GEZGIN is managed by the CCIC. In normal operation the CCIC enables reception of multi-spectral images on four channels into the FPGA for wavelet filtering and transmission of compressed images through the Buffered Serial Port (BSP) of the DSP to the primary SSDR unit.

In case of FPGA failure, the CCIC can directly route a single band image into the DSP for gracefully degraded operation. In this case, wavelet transformation is performed by the DSP.

In case of failure of the primary SSDR unit, the CCIC is capable of routing the compressed images to the secondary SSDR through ÇOBAN payload. The GEZGIN ÇOBAN link is two way allowing ÇOBAN to transfer its image output to the primary SSDR through GEZGIN.

Command and Control Interface

The command and control flow into GEZGIN is through a Controller Area Network (CAN)[6]. GEZGIN is connected to two CAN buses, running in cold redundancy. The CAN interface is supervised by a dedicated micro-controller with an embedded CAN controller (C515), which is connected to the Serial Port Interface (SPI) of the DSP. The On-Board Computer (OBC) on BILSAT issues commands through the CAN interface to activate, boot, deactivate and configure GEZGIN.

The OBC can reboot/reconfigure the FPGA or the DSP on GEZGIN independently in cases of radiation induced malfunctions or software upgrades. In case of upgrades, new program code of the DSP or new configuration code of the FPGA is received through the CAN bus and transferred to the related device. In case a code upgrade cannot be received within one mission cycle of GEZGIN (due to limited bandwidth on the CAN bus), the code is transferred to a secondary Flash RAM for temporary storage allowing reception spanning multiple cycles. A permanent copy of all program/configuration codes remains intact for fail-safe operation.

Table 1. Performance results of the JPEG2000 algorithms implemented by GEZGIN.

	Size, PSNR					
Image (Size)	Lossless	Lossy				ROI + T4L
		T1L	T2L	HH	HH+T1L	
ERC_TM_512 (786432 bytes)	546099 bytes, ∞ dB	444652 bytes, 41.25 dB	345006 bytes, 32.98 dB	417194 bytes, 38.15 dB	340294 bytes, 36.55 dB	362291 bytes, 29.43 dB
DENVER512 (786432 bytes)	627260 bytes, ∞ dB	523476 bytes, 41.21 dB	423646 bytes, 32.88 dB	479154 bytes, 29.76 dB	401310 bytes, 29.49 dB	368262 bytes, 28.08 dB
LENA256 (196608 bytes)	113688 bytes, ∞ dB	87945 bytes, 41.34 dB	64902 bytes, 32.76 dB	88419 bytes, 40.29 dB	69020 bytes, 38.00 dB	49620 bytes, 27.28 dB

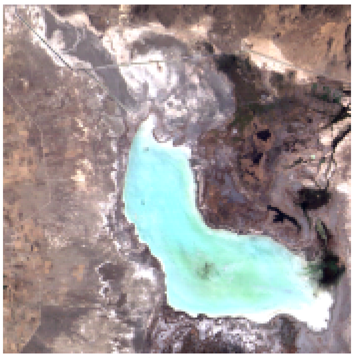
T1L : Truncation of 1 LSB of Wavelet Coefficients

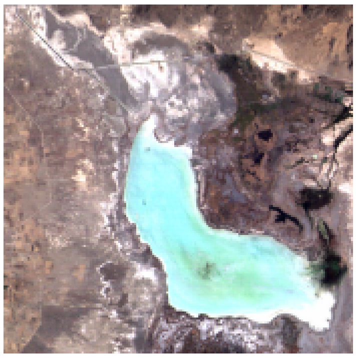
T2L : Truncation of 2 LSB of Wavelet Coefficients

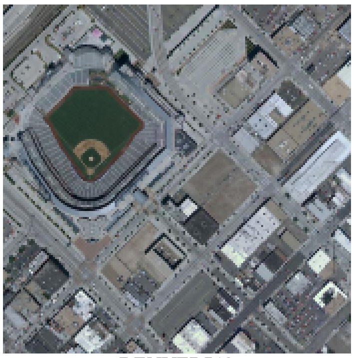
T4L : Truncation of 4 LSB of Wavelet Coefficients

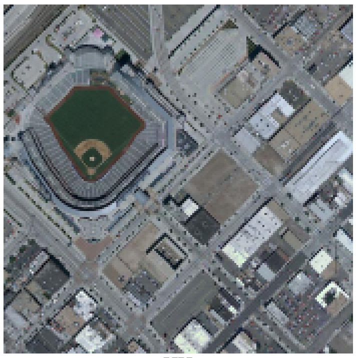
HH : Omission of HH band at level 1

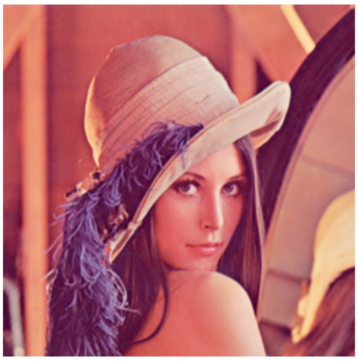
HH + T1L : Omission of HH band at level 1 + Truncation of 1 LSB of Wavelet Coefficients


ERC_TM_512


T1L


DENVER512


HH


LENA256


ROI+T4L (ROI is around the eyes)

Figure 9. Samples of original and compressed images.

High-speed design considerations

The operational frequencies of processing units on GEZGIN give rise to signal integrity issues related to clock management and bus design. The built-in PLL units of the FPGA and DSP allow routing of one fourth of the core frequencies at the board level. Minimal clock skew and glitch free clock signals are attained using a source terminated point-to-point distribution. Address and high-speed control signals are also distributed like clock signals. Signal integrity on the data bus is maintained by dynamically separating the bus with switches and creating point-to-point connections between transmitting and receiving units.

Conclusion

In this paper, GEZGIN, a highly flexible and reconfigurable signal processing payload employing JPEG2000 compression algorithm is introduced. GEZGIN attains a significant increase in the data storage, down-link capacity and robustness in noisy channels, while maintaining cost-effectiveness by using COTS.

GEZGIN uses a high degree of parallelism among image processing units. The compute intensive tasks of JPEG2000 algorithm are implemented in dedicated hardware to maintain real-time operation. The proposed hardware allows for full reconfigurability of all processing units, resulting in a wide selection of operating modes, in-orbit program upgrade possibilities, and flexibility for addition of new image processing features. Failsafe operation is guaranteed through the usage of redundant hardware.

References

1. A. Bradford, L. Gomes, G. Yüksel, C. Özkaptan, "BILSAT-1: A Low-cost, Agile, Earth Observation Micro-satellite for Turkey", IAF2002, October 2002.
2. ISO/IEC JTC 1/SC 29/WG 1 N1646R, "JPEG2000 Image Coding System"
3. A. Brewer, "Implementation of Image Compression on PoSAT-1", International Journal of Small Satellite Engineering, Issue 1, October 1995.
4. P. Hou, M. Petrou, C. Underwood, A. Hojjatoleslami, "Improved JPEG performance in conjunction with cloud editing for Remote Sensing applications", IEEE Transactions on Geoscience and Remote Sensing, January, 1999, pp. 515-524.
5. P.S. Yeh, J. Venbrux, P. Bhatia, W. H. Miller, "A Visually Lossless Data Compression Technique for Real-Time Frame/Pushbroom Space Science Imagers", Applications of Digital Image Processing XXII, Proceedings of SPIE Vol. 4115, December 2000, p. 320-331.
6. W. Wijmans, P. Armbruster, "Data Compression Techniques for Space Applications", Review of Current ESA/ESTEC Development, Announcement of a Round Table", DASIA'96, May 1996, Rome.
7. TIA/EIA-644, "Electrical Characteristics Of Low Voltage Differential Signaling (LVDS) Interface Circuits"
8. C. Chrysafis and A. Ortega, "Line Based Reduced Memory Wavelet Image Compression." IEEE Transactions on Image Processing, March 2000, Vol 9, No 3, pp. 378-389.
9. I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps." Journal of Fourier Anal. Appl. , 1998, Vol. 4, Nr. 3, pp. 247-269.
10. J. Reichel, M. J. Nadenau, M. Kunt, "Row-based wavelet decomposition using the lifting scheme." Proceedings of the Workshop on Transforms and Filter Banks (WTFB'99), Brandenburg an der Havel, Germany, March 5-7, 1999.
11. D. S. Taubman, M. W. Marcelin, "JPEG2000 Image Compression Fundamentals, Standards and Practice", Kluwer Academic Publishers, 2002.
12. J. M. Shapiro, "Embedded image coding using zero-trees of wavelet coefficients," IEEE Trans. on Signal Processing, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
13. A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 243–250, June 1996.
14. CAN Specification 2.0B, http://www.can.bosch.com/content/literature.html